

Remarks

The restriction requirement

Applicants have hereby elected Group I for examination without traverse and have canceled the
5 claims of Group II without prejudice, thereby completely satisfying Examiner's restriction requirement of 2/12/02.

The amendments to claims 24, 57 and 89

10 Examiner will immediately see that these amendments merely deal with typographical matters and neither add new matter nor materially change the scope of the claims.

New claims 112-131

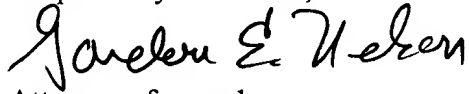
New claims 112-131 are fully supported by the disclosure of the present application and
15 consequently add no new matter. Taking claim 112 as an example, the "request" of the claim is embodied in the queries of the present application, the "plurality of database systems" is at least embodied in cache database 347 and source database 241, with cache database 347 embodying the "first database system" and source database 241 embodying the "second database system". The "redirector" is embodied in DA interface 304 and query dispatcher 351. The manner in which these
20 components of this embodiment of the invention interact to "respond[]" to the request when the request includes a specifier that cannot be interpreted in the first database system by causing the request to be executed at least in part in a second database system of the plurality, the request otherwise being executed in the first database system" is described at least at page 12, lines 15-24.

25 Given the history of restrictions in the present application, Applicants' attorney believes that it is worthwhile pointing out that new claims 112-131 are generic to the claims remaining in the application after the restriction and election, and that consequently, the addition of claims 112-131 should not result in a further restriction requirement.

Conclusion

Applicants have been fully responsive to Examiner's restriction requirement and have demonstrated that neither the amendments to claims 24, 57, and 89 nor new claims 112-131 add new matter. Since the total number of claims presently in the application is less than the 105 claims for which fees have already been paid, no fees are believed to be required by way of this election. Should any be required, please charge them to deposit account number 50315.

Respectfully submitted,



Attorney of record,

Gordon E. Nelson

57 Central St., P.O. Box 782

Rowley, MA, 01969,

Registration number 30,093

Voice: 1-978-948-7632

Fax: 1-978-878-0156

3/12/02

Date

1 **24.** (twice amended) An improved server of the type that provides a program with a standard
2 interface for querying remote datasets,
3 the [improvement] improved server comprising:
4 a queryable cache that contains copies of certain of the datasets and is local to the server,
5 the improved server receiving a query for a remote dataset in a form required by the interface from
6 the program, determining whether a copy of a dataset to be queried is present in the queryable cache,
7 and, if the copy is present, querying the copy, and otherwise querying the remote dataset,
8 whereby the queryable cache is transparent to the program.

1 **25.** (amended) The improved server set forth in claim 24 wherein
2 the program uses global identifiers for the remote data sets and
3 the copies in the queryable cache have local identifiers; and
4 the improved server further comprises:
5 a query analyzer that receives the global identifier for a dataset being queried and if there is a
6 copy of the data set indicated by the global identifier, returns the local identifier to the server,
7 the server using the local identifier to query the copy.

1 **26.** (amended) The improved server set forth in claim 25 wherein:
2 the query analyzer further indicates to the server whether the copy of the dataset is in the
3 queryable cache.

1 **27.** (amended) The improved server set forth in claim 24 further comprising:

2 a dataset manager that determines a dataset for which a copy is needed in the cache,
3 obtains a copy of the remote dataset, and adds the copy to the cache.

1 **28.** (amended) The improved server set forth in claim 27 wherein:

2 the dataset manager further determines a dataset for which a copy is no longer needed in
3 the cache and removes the copy from the cache.

1 **29.** (amended) The improved server set forth in any of claims 27 or 28 wherein:

2 the dataset manager determines whether to add or remove a dataset by determining a
3 likelihood that a query will be made to the dataset.

1 **30.** (amended) The improved server set forth in claim 29 wherein the improved server further
2 comprises:

3 a query log that lists past queries that have been made to the standard interface and
4 the dataset manager uses the query log to determine a likelihood that a query will be made to
5 a dataset.

1 **31.** (amended) The improved server set forth in claim 29 wherein:

2 the dataset manager uses information about an event that will result in queries to a
3 dataset to determine a likelihood that a query will be made to a dataset.

1 **32.** (amended) The improved server set forth in claim 31 wherein:

2 the dataset manager uses information about a time of occurrence of the event to
3 determine a likelihood that a query will be made to a dataset.

1 **33.** (amended) The improved server set forth in claim 24 wherein

2 when a change occurs in a remote dataset of the remote datasets, an indication including
3 the change is sent to the server without intervention by the server and

4 the improved server further comprises:

5 an update receiver that receives the indication and modifies any copy of the changed dataset
6 as required by the indication.

1 **34.** (amended) A method of querying datasets in a server that provides a standard

2 interface for querying remote data sets to a program executing on the server

3 the method comprising the steps of:

4 receiving a query for a remote dataset in a form required by the standard interface;

5 determining whether a copy of a dataset to be queried is present in a queryable cache
6 local to the server; and

7 if the copy is present in the queryable cache, querying the copy and otherwise
8 querying the remote dataset,

9 whereby the queryable cache is transparent to the program.

1

1 **35.** The method set forth in claim 34 wherein

2 the form required by the standard interface uses global identifiers for the remote
3 data sets and

4 the copies in the queryable cache have local identifiers; and

5 the method further includes the steps of :

6 providing the global identifier for a dataset being queried to a query analyzer in the
7 server; and

8 if there is a copy of the data set indicated by the global identifier, receiving the local
9 identifier from the query analyzer,

10 the local identifier being used in the step of querying the local copy.

1

1 **53.** The method set forth in claim 35 further comprising the step of:

2 receiving an indication from the query analyzer whether the copy is present in the
3 queryable cache.

1 **54.** The method set forth in claim 34 further comprising the steps of:

2 determining a dataset for which a copy is needed in the cache;

3 obtaining the copy; and

4 adding the copy to the cache.

1 **55.** The method set forth in claim 54 further comprising the steps of:

2 determining a dataset for which a copy is no longer needed in the cache; and

3 removing the copy from the cache.

1 **56.** The method set forth in any one of claims 27 or 28 wherein:

2 the step of determining a dataset is performed by determining a likelihood that a
3 query will be made to the dataset.

1 **57.** (amended) The method set forth in claim 56 wherein:

2 in the step of determining a dataset, a query log that lists [pasts] past queries is used
3 to determine the likelihood that a query will be made.

1 **58.** The method set forth in claim 56 wherein:

2 in the step of determining a dataset, information about an event that will result in
3 queries to a dataset is used to determine the likelihood that a query will be made.

1 **59.** The method set forth in claim 58 wherein:

2 in the step of determining a dataset, information about a time of occurrence of the
3 event is used to determine the likelihood that a query will be made.

1 **60.** The method set forth in claim 34 wherein

2 when a change occurs in a remote dataset of the remote datasets, an indication
3 including the change is sent to the server without intervention by the server and

4 the method further comprises the steps of:

5 receiving the indication and modifying any copy of the changed dataset as required by
6 the indication.

1 **84.** A memory device characterized in that:
2 the memory device contains code which, when executed by a processor, performs a method
3 of querying datasets in a server that provides a standard interface for querying remote data sets to a
4 program executing on the server,
5 the method comprising the steps of
6 receiving a query for a remote dataset in a form required by the standard interface;
7 determining whether a copy of a dataset to be queried is present in a queryable cache local to
8 the server; and
9 if the copy is present in the queryable cache, querying the copy and otherwise querying the
10 remote dataset,
11 whereby the queryable cache is transparent to the program.

1 **85.** The memory device set forth in claim 84 further characterized in that:
2 in the method, the form required by the standard interface uses global identifiers
3 for the remote data sets and
4 the copies in the queryable cache have local identifiers; and
5 the method further includes the steps of :
6 providing the global identifier for a dataset being queried to a query analyzer in the
7 server; and
8 if there is a copy of the data set indicated by the global identifier, receiving the local
9 identifier from the query analyzer,

10 the local identifier being used in the step of querying the local copy.

1

1 **86.** The memory device set forth in claim 84 wherein the method further comprises the
2 steps of:

3 determining a dataset for which a copy is needed in the cache;

4 obtaining the copy; and

5 adding the copy to the cache.

1 **87.** The memory device set forth in claim 86 wherein the method further comprises the
2 steps of:

3 determining a dataset for which a copy is no longer needed in the cache; and

4 removing the copy from the cache.

1 **88.** The memory device set forth in any one of claims 86 or 87 wherein:

2 the step of determining a dataset is performed by determining a likelihood that a
3 query will be made to the dataset.

1 **89.** (amended) The memory device set forth in claim 88 wherein:

2 in the step of determining a dataset, a query log that lists [pasts] past queries is used
3 to determine the likelihood that a query will be made.

1 **90.** The memory device set forth in claim 88 wherein:

2 in the step of determining a dataset, information about an event that will result in

3 queries to a dataset is used to determine the likelihood that a query will be made.

1 **91.** The memory device set forth in claim 90 wherein:

2 in the step of determining a dataset, information about a time of occurrence of the
3 event is used to determine the likelihood that a query will be made.

4

1 **92.** The memory device set forth in claim 84 wherein

2 when a change occurs in a remote dataset of the remote datasets, an indication
3 including the change is sent to the server without intervention by the server and
4 the method further comprises the steps of:

5 receiving the indication and modifying any copy of the changed dataset as required by
6 the indication.

1 **93.** The memory device set forth in claim 85 wherein the method further comprises the step
2 of:

3 receiving an indication from the query analyzer whether the copy is present in the
4 queryable cache.

1 **112.** Apparatus for responding to a request, the request including one or more specifiers
2 referring to objects belonging to a plurality thereof in a distributed database system that
3 includes a plurality of database systems and
4 the apparatus comprising:
5 a first database system of the plurality; and

6 _____ a redirector which responds to the request when the request includes a specifier that
7 cannot be interpreted in the first database system by causing the request to be executed at
8 least in part in a second database system of the plurality, the request otherwise being
9 executed in the first database system.

1 **113.** The apparatus set forth in claim 112 wherein:
2 _____ the objects in the first database system include copies of objects contained in
3 at least one other database system belonging to the distributed database system.

1 **114.** The apparatus set forth in claim 113 wherein:
2 _____ the first database system functions as a cache with regard to the objects whose copies
3 are included therein.

1 **115.** The apparatus set forth in claim 113 wherein the other database system is the second
2 database system.

1 **116.** The apparatus set forth in claim 115 wherein:
2 _____ the first database system functions as a cache with regard to the second database
3 system.

1 **117.** The apparatus set forth in any one of claims 112 through 116 wherein:
2 _____ the apparatus is local to a server of the type that provides a program executing in the
3 server with a standard interface for querying databases; and
4 _____ the requests include queries received via the standard interface.

1 118. The apparatus set forth in claim 117 wherein:

2 the server obeys the http protocol and the program is a Web application program.

1 119. A method of responding to a request, the request including one or more specifiers that
2 refer to objects belonging to a plurality thereof in a distributed database system that includes
3 a plurality of database systems and

4 the method comprising the steps of:

5 receiving the request in a first database system of the plurality;

6 determining whether the request includes a specifier that cannot be interpreted in the
7 first database system of the plurality; and

8 when the request includes such a specifier, causing the request to be executed at least
9 in part in a second database system of the plurality.

1 120. The method set forth in claim 119 wherein:

2 the objects in the first database system include copies of objects contained in at least

3 one other database system belonging to the distributed database system,

4 whereby the first database system functions as a cache with regard to the objects whose
5 copies are included therein.

1 121. The method set forth in claim 120 wherein:

2 the other database system is the second database system,

3 whereby the first database system functions as a cache with regard to the second database
4 system.

1 **122.** The method set forth in any one of claims 119 through 121 wherein:
2 the first database system is local to a server of the type that provides a program
3 executing in the server with a standard interface for querying databases; and
4 in the step of receiving the request, the request is received via the standard interface.

1 **123.** The method set forth in claim 122 wherein:
2 the server obeys the http protocol and the program is a Web application program.

1 **124.** A memory device characterized in that:
2 the memory device contains code which, when executed in a processor, performs a
3 method of responding to a request, the request including one or more specifiers that refer to
4 objects belonging to a plurality thereof in a distributed database system that includes a
5 plurality of database systems and
6 the method comprising the steps of:
7 receiving the request in a first database system of the plurality;
8 determining whether the request includes a specifier that cannot be interpreted in the
9 first database system of the plurality; and
10 when the request includes such a specifier, causing the request to be executed at least
11 in part in a second database system of the plurality.

1 125. Apparatus for caching copies of objects belonging to a subset of the objects belonging
2 to a first database system that returns an object in response to a request therefor, the request
3 including one or more specifiers referring to the objects and
4 the apparatus comprising:
5 a second database system that contains the copies; and
6 a redirector that responds to the request when the request includes a specifier that
7 cannot be interpreted in the second database system by causing the request to be executed at
8 least in part in the first database system, the request otherwise being executed in the second
9 database system.

1 126. The apparatus set forth in claim 125 wherein:
2 the apparatus is local to a server of the type that provides a program executing in the
3 server with a standard interface for querying databases; and
4 the requests include queries received via the standard interface.

1 127. The apparatus set forth in claim 126 wherein:
2 the server obeys the http protocol and the program is a Web application program.

1 128. A method of responding to a request that includes one or more specifiers referring to
2 objects belonging to a set thereof where the objects are stored in a first database system and
3 copies of a subset thereof are stored in a second database system,
4 the method comprising the steps of:

5 receiving the request in the second database system;
6 determining whether the request includes a specifier that cannot be interpreted in the
7 second database system; and
8 when the request includes such a specifier, causing the request to be executed at least
9 in part in the first database system instead of in the second database system.

1 129. The method set forth in claim 128 wherein:

2 the second database system is local to a server of the type that provides a program
3 executing in the server with a standard interface for querying databases; and
4 in the step of receiving the request, the request is received via the standard interface.

1 130. The method set forth in claim 129 wherein:

2 the server obeys the http protocol and the program is a Web application program.

1 131. A memory device characterized in that:

2 the memory device contains code which, when executed in a processor, performs a
3 method of responding to a request that includes one or more specifiers referring to objects
4 belonging to a set thereof where the objects are stored in a first database system and copies of
5 a subset thereof are stored in a second database system,
6 the method comprising the steps of:
7 receiving the request in the second database system;
8 determining whether the request includes a specifier that cannot be interpreted in the

- 9 second database system; and
- 10 _____ when the request includes such a specifier, causing the request to be executed at least
- 11 in part in the first database system instead of in the second database system.